FIG. 1: Data Packet Transport System

Switching
Equipment 115

Switching
Equipment 115

Synchronized
Packet-Based
Trunk 110

Synchronized
Packet-Based
Trunk 110

Communications
Ring 101

1
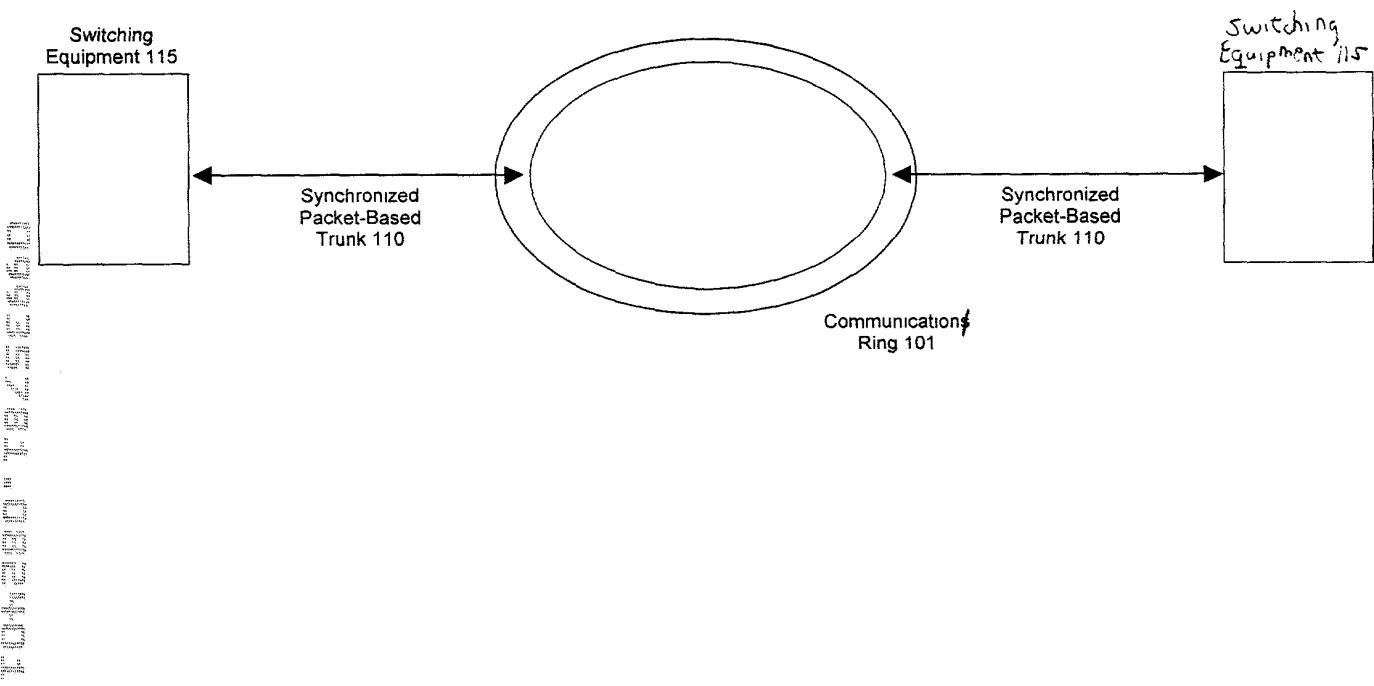
FIG. 2: Timing Diagram of Synchronized Packet-based Trunk

Communication Channel 201

Frame cycle 205

Frame cycle 205

Frame Synchronization Packet 210

TDM Packets 215

Asynchronous Data Packet 220

Frame Synchronization Packet 210

TDM Packets 215

Asynchronous Data Packet 220

. . .
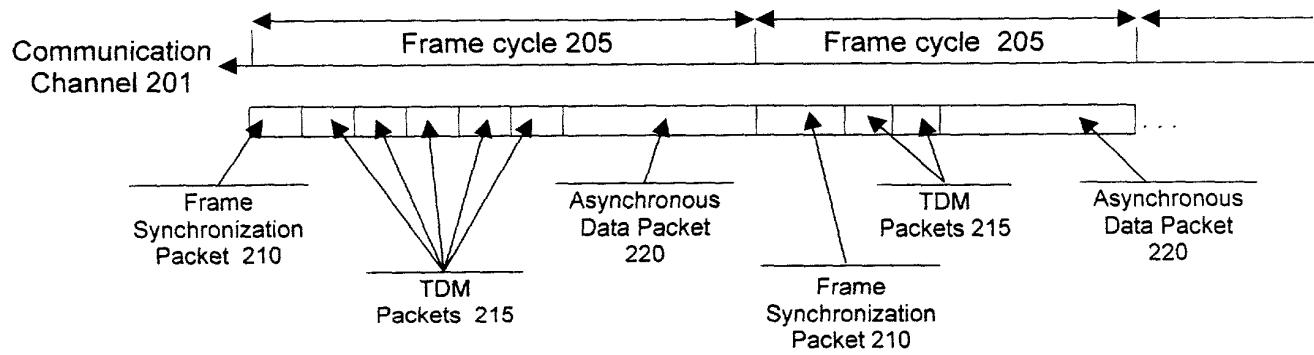
**FIG. 3: Trunk Transmitter and Receiver**

FIG. 4: Frame Synchronization Packet Format

401	1	1	1	1	0	1	1	0	F1
402	1	0	1	0	1	0	0	0	F2

403	Byte-1
404	Byte-2
405	Byte-3
406	Byte-4
407	OAMP bytes

Fixed-bytes of Payload e.g 14 bytes

FIG. 5: TDM Packet Format

501	Byte-1
502	Byte-2
503	TDM payload
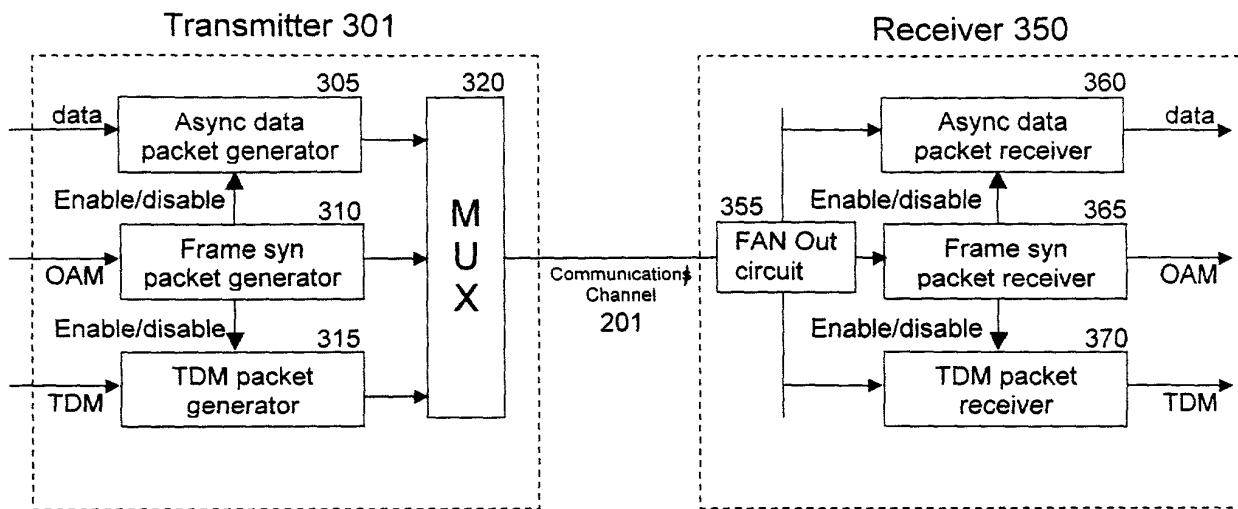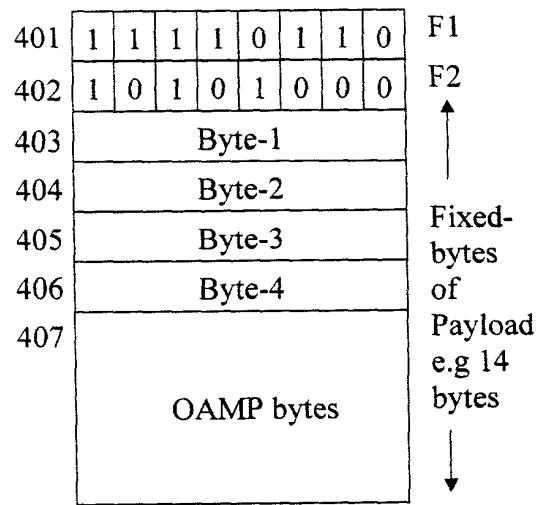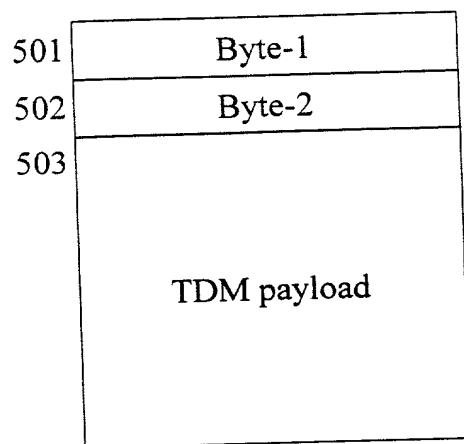
FIG. 6: Asynchronous Data Packet Format

| 601 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

HDLC Flag

602

Asynchronous Data Bytes

Variable length

| 603 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

HDLC Flag

6
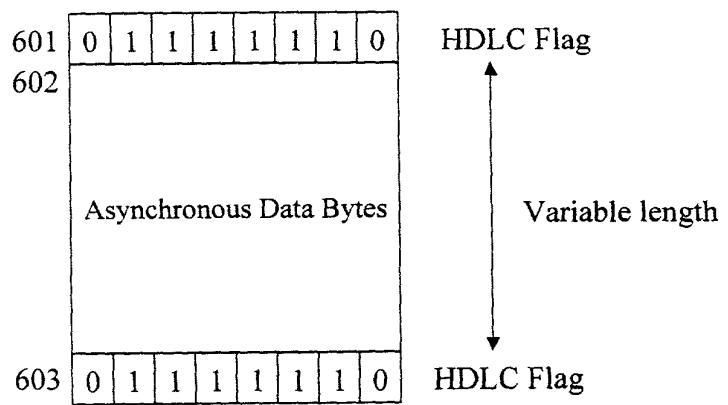
FIG. 7: Synchronized Packet-Based Trunk Receiver Operation

701

Bit stream
from
Communications
Channel

Frame

Frame
detected

705

Payload De-
Scrambler
and Sync
Packet
Processing

711

Last TDM
Packet?

no

710

TDM Packet
Processor

715   yes

Data
Packet
Processor

720

yes

no

Correct
frame pattern
?

FIG. 8: Data Packet Processor

Data Packet
Processor 715

711
Last TDM Packet? — yes → 801 Partial packet from previous frame? — yes → 808 Continue the partial data payload from previous 125 us frame

720
Correct frame pattern? — yes / no

no → 802 End of a frame cycle? — yes → Correct frame pattern

no → 803 Opening Flag? — no

yes → 804 Ending Flag? — yes → 809 Store one data packet

no → 805 Store one data byte

no → 806 End of a frame cycle?

yes → 807 Store the partial data payload for continuation in the next 125 us frame

8

FIG. 9: Synchronized Packet-Based Trunk Transmitter ~~Operation~~

Pointer Directory 901

Frame sync signal

Directory Processor 915

Pointer to the location of next TDM or data packet byte

Partial Packet Counter 930

Input bit stream

Input Processor 910

Shared Memory 905

Communication Register 920

Output Processor 925

To Communications Channel

Output bit stream

9

FIG. 10: Example of Pointer Directory Implementation

901
Pointer Directory

8-byte pointer

Pointer directory can be organized
Based on priority code

1st priority
Sub-directory

2nd priority
Sub-directory

3rd priority
Sub-directory

1010

8-byte pointer

Status 1	1011
Status 2	1012
Pointer Byte 1	1013
Pointer Byte 2	1014
Pointer Byte 3	1015
Data Length Byte 1	1016
Date Length Byte 2	1017
Date Length Byte 3	1018

Active/
Inactive

Status1   1011

Priority codes

Status1   1012

Priority codes

After a TDM or data packet is sent, the pointer
for that packet is erased and the Active/Inactive bit
is set to be inactive

10

FIG. 11: Relationship between Pointer Directory and Shared Memory

901
Pointer Directory

Shared Memory   905

FIG. 12: Synchronized Packet-Based Input Processor

Trunk ↑

Input Processor 910

Start
1201
Packet Received?

Yes 1202

Available
Memory space?

No → Generate ~~congestion~~ memory unavailable signal 1205

Yes 1203

Create
Subdirectory based on
Priority

1206
Drop the packet

1204
Write packet
Into Shared Memory

12

FIG. 13: Synchronized Packet-Based Trunk Directory Processor

Directory Processor   915

Start → **Line up with the frame Sync signal**   1301

→ Yes   1302
Set the communication register to the pointer to sync packet and then TDM packet

no → **End of last TDM packet?**   1303

Incomplete data packet left from previous frame cycle   1304   yes

no   1305
Frame Sync Detected?   yes

no   1306
Set the communication register the next packet to be sent out based on priority

Read the partial packet counter and set it to the comm register   1307

yes → **Complete the data packet before frame sync**   1308

no

Incomplete data packet?   1309

no   yes   1310

Record the address of the next byte of the incomplete data packet into a partial packet counter

13

FIG. 14: Synchronized Packet-Based Trunk Output Processor

Output Processor   925

Start

1401

Line up with the frame sync signal

1402

Read packets out of shared memory based on the value of comm register pointer

1403

Line up with Frame Sync signal?

yes

no

FIG. 15: Communication Ring

Node 1510

Communication Ring
1501

Node 1510

Node 1510

Direction
of
Traffic Flow

Over-Subscription Traffic 1520

Subscription Traffic 1515

Node 1510

Node 1510
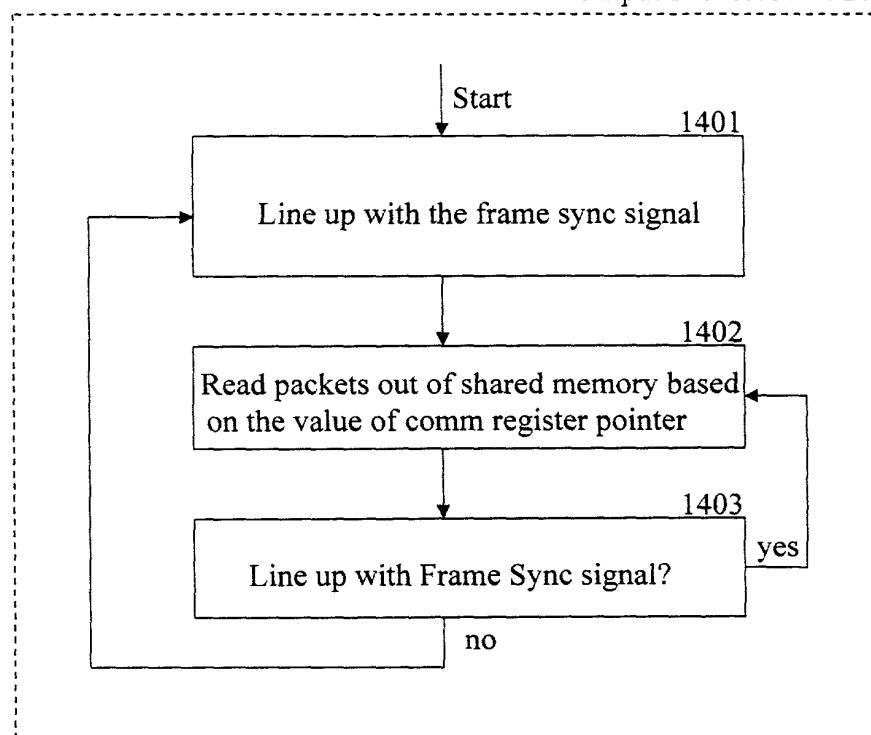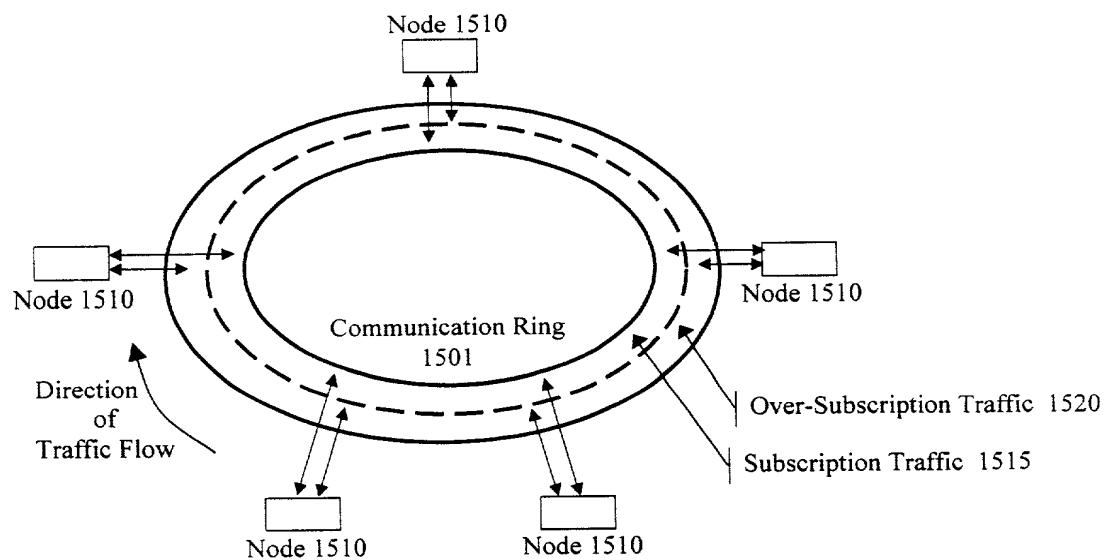
FIG. 16 Traffic Processing Stages

Input traffic → **Stage 1 1601** Traffic Identification → **Stage 2 1602** Traffic Conformance/ Policing → **Stage 3 1603** Egress Queuing, Scheduling, RED → Output traffic Communication Ring 1501
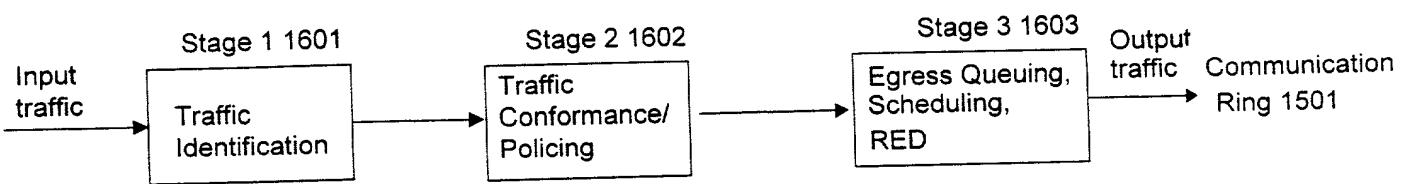
**FIG. 17 Dynamic Bandwidth Sharing Traffic Processing**
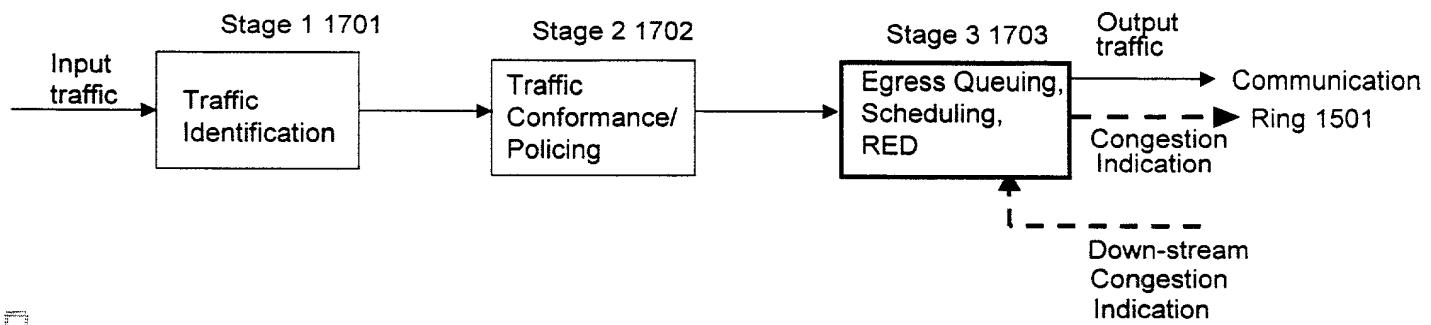
FIG. 18 Dynamic Bandwidth Sharing Traffic Processing Functional Diagram
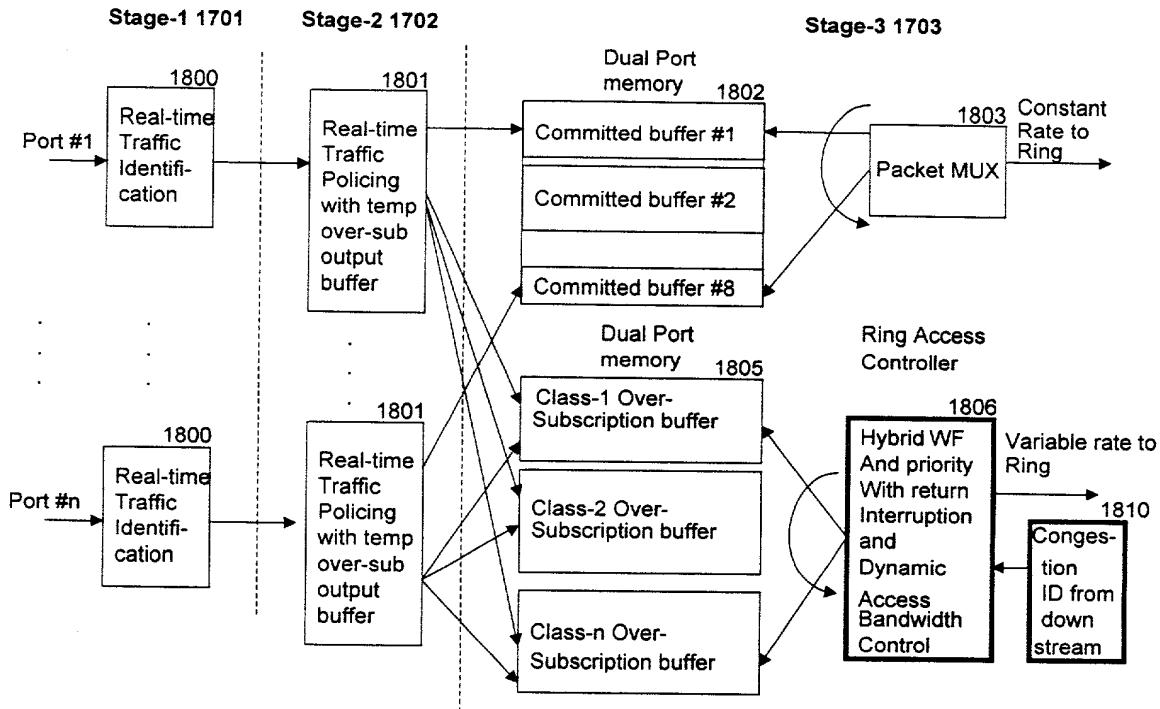
FIG. 19 Example of Traffic Identification Table

Traffic Identification Table 1901

Ethernet port #2	Based on per port contract VBR- nrt traffic, mean rate 2 Mb/s, Normal burst size 500 bytes, Excess burst size 1500 bytes
Ethernet port #4	Based on per DSCP contract TOSxxx : VBR- nrt . mean rate 4 Mb/s Normal burst size 128 bytes Excess burst size: 500 bytes TOSyyy : PCR, mean rate 64 Kb/s, Delay variation 50 us

FIG. 20 Dynamic Bandwidth Sharing Stage 2 Processing

Stage-2 1702

20

FIG. 21 Dynamic Bandwidth Sharing Stage 3 Processing

Committed Traffic 2101

1801

Real-time Traffic policing

PCR, SCR

Dual Port memory 1802

Committed buffer #1

Committed buffer #2

Committed buffer #8

1803

Constant Rate to Ring

Packet MUX

21

FIG. 22 Dynamic Bandwidth Sharing Stage 3 Processing

Over-Subscription Traffic 2201

Dual Port
memory   1805

Class-1 Over-
Subscription buffer

Be #1

Be #n

Ring Access
Controller

1806

Class-2 Over-
Subscription buffer

Be #i

RR

Acceptable
over-sub
parameter

Max.
Over-sub
parameter

1810

Congestion
ID from
down
stream

Class-n Over-
Subscription buffer

Access
Bandwidth
meter

Variable rate
To Ring

parameter

FIG. 23 Dynamic Bandwidth Sharing Congestion Control

Ring Access Controller 1806

Dual Port
memory   1805

Access Bandwidth

2305

| Class-1 Over-Subscription buffer |
| Be #1 |
| Be #n |
| |
| Class-2 Over-Subscription buffer |
| Be #i |
| |
| Class-n Over-Subscription buffer |

~~Congested over sub meter~~
Adjustment processor

Bandwidth available on the ring

2401

RR

SCHEDULER

Transmit data
To the ring

WF Scheduler table 2402

Class-1	# of bytes
Class-2	# of bytes
Class-n	# of bytes

23

FIG. 24 Dynamic Bandwidth Sharing

```
                1803    Constant
┌──────────────┐        Rate  to
│              │        Ring
│  Packet MUX  │                      Communication
│              │
│              │                      **Ring 1501**
└──────────────┘
                          ┌────────────────────────────────┐
                          │▨▨▨▨▨▨▨▨│  ████████  │           │
                          └────────────────────────────────┘

                          ┌────────────────────────────────┐
Ring access               │▨▨▨▨▨▨▨▨│    ██████  │           │
Controller                └────────────────────────────────┘

                          ┌────────────────────────────────┐
            1806          │▨▨▨▨▨▨▨▨│   ██████   │           │
┌──────────────┐          └────────────────────────────────┘
│ Hybrid WF    │   Variable rate to
│ And priority │   Ring
│ With return  │
│ Interruption │
│    and       │                  1810
│ Dynamic      │          ┌─────────────────┐
│ Access       │◄─────────│ Congestion      │
│ Bandwidth    │          │ ID from down    │
│ Control      │          │  stream         │
└──────────────┘          └─────────────────┘
```